UNIVERSITY OF DAR ES SALAAM

COLLEGE OF INFORMATION AND COMMUNICATION TECHNOLOGIES (COICT)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PRACTICAL TRAINING REPORT

REPORT TITLE: DEVELOPMENT OF FLIGHT RESERVATION SYSTEM

Student Name:	Mahende Chacha Maro
Reg. No. :	2018-04-06802
Practical Training Year:	PT2
Company/organization name:	ICT PACK SOLUTIONS LTD
Training Officer´s Name:	Mr.Denis Frimos Mlewe
Academic Supervisor's Name:	Mr.Kennedy Mwakisole

DECLARATION

I MAHENDE, Chacha Maro declare that, this report is my own work and has not been submitted or presented to any other Institute or University for a similar or any other degree award.

Student’s Name: Mahende Chacha Maro   Training Officer: Denis Frimos Mlewe

Signature: ……………........................   Signature………………………

Date…………………………   Date……………………………

ACKNOWLEDGEMENT

Firstly, I thank God for His blessing and guidance since I started up until the end of this practical Training. I would like to give thanks to management of **ICT PACK** for giving me an opportunity to conduct my field practical at their company thus not only help me to increase my knowledge in IT field but also help me to gain experience on how the real working environment is and it has really been a rewarding experience.

Special thanks to my Training Supervisor Mr.Denis Frimos for his guidance and providing a friendly environment for my study field.

Furthermore, many thanks to all ICT PACK Staff for their contributions to my field from date I started up to the end, especially Mr.Innocent, Mr.Alex, Mr.Stebins and Mr.Vitus for their kindness, training, and team work they had with me.

Lastly Special thanks are conveyed to the management of the University of Dar es Salaam (UDSM) for being so helpful in imparting me with knowledge and awareness on my Practical Training, much thanks to my classmates and friends who have played a big part in imparting knowledge in me through idea exchanging and giving advice.

ABSTRACT

The contents in this report highlights what has been observed and done during the whole fieldwork program as it covers the period from **14th** September up to **23rd** October **2020**. The fieldwork was conducted in ICT PACK SOLUTIONS LTD, Dar es- salaam. This report is based in Web Development. During my Practical Training I achieved a skill of listening to customers, Analyzing and understanding what a Customer needs, Improved my Skill of Interface design.

The field report is divided into two chapters, chapter one contains an Introduction to my We Application and Technologies used in developing the web Application. Chapter two demonstrates how the web application works, The Project Structure, The functionalities implemented, The Database Design and All the Interfaces.

DEDICATION

This report is a special Dedication to the University of Dar es Salaam for ensuring and proving quality education theoretically and practically through practical Training.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

COICT	College of Information Communication and Technology
IT	Information Technology
UML	Unified modeling language
CRUD	Create read update delete
PT	Practical training
BSC	Bachelor of Science
JPA	Java Persistence API
API	Application Programming Interface
MVC	Model, View And Controller

LIST OF FIGURES

CHAPTER ONE

1.1 INTRODUCTION

The main objective is to develop system that helps passengers/travelers who use Airplanes as means of transport to easily book/reserve their flights through the website. The development of this website is done on the Spring MVC. Spring MVC is The Java Framework which is used to build Web Applications. It follows the Model-View-Controller design pattern. It implements all the basic features of a core **spring** framework like Inversion of Control and Dependency Injection.

1.2 PROBLEM IDENTIFICATION

A lot of people move to offices so as to book/reserve tickets for flights, hence use a lot of time and cost. Also high possibility of errors occurs due to the use of manual system of reservation and boarding process.

Due to this paper based system it has resulted to inability of passengers to select their preferred seat(s) from the reservation, no option of passengers printing their boarding pass, non-notification of passengers of flight cancellation or delays and passengers don't have access to aircraft maintenance report to ease the fears associated with the air travel and its disasters.

1.3 PROPOSED SOLUTION ON SOLVING MAIN PROBLEM

For small organization a paper based approach is adequate, however for large organizations, an automated approach is desirable. An automated approach to reserving and scheduling flights would eliminate the errors described. A web application will help people to book/reserve flights at any time and place they are, without going to the booking office.

Such an automated solution would need to incorporate the following features in order to solve to solve the problem:

- Reservation can be easily cancelled, thereby encouraging clients to cancel "just in case" reservations.
- Once a seat is reserved, the new reservation visible will contain a minimum seat compared to the previous seats.
- Clients can be automatically informed of changes to a reservation.

1.4 JUSTIFICATION OF SOLUTIONS

Flight Reservation System is the appropriate solution to solve the problem as it is cheap and clients will be able to access the system at any place. Not only that it is the right solution to implement because it is secure, user-friendly and has an attractive user interface.

The system can be able to address a wide range of problems since it is multipurpose, consist of multiple functionalities and can be easily implemented or accomplished successful at a short time with a high impact.

Lastly, it can run into different platform or electronic devices because it is a web Application making it the best solution fit to address the problem.

1.4.1 FLIGHT RESERVATION SYSTEM

Programming language used is java with version 11, Html, CSS in order to develop a solution of the problem. Frontend (user interface) has been developed with Html, CSS and Thymeleaf Template Engine and Backend I used Java in framework called SpringBoot which has some dependencies that contributed in development of the project.

The following are dependencies used from spring boot

- Spring Security
- Spring Web
- Spring Data Jpa
- spring Devs Tools
- Spring boot Starter Mail
- Itext pdf

1.5 FUNCTIONAL FUNCTIONALITIES

The system developed is going to have the following functionalities as the user (Administrator) will login then he will be able to

- Create
- read
- update and
- delete information(CRUD)

Apart from that also the user (Administrator) will be able to activate and deactivate users to use the system due to various reason.

1.6 NON FUNCTIONAL FUNCTIONALITIES

One of non-functional part of the system is Security

UML USE CASE DIAGRAM

Uml use case diagram describes how a user uses a system to accomplish a particular task. The diagram below shows how the user uses the system and how the Admin perform certain functionalities after Signing in the System. In order to perform some of system functionalities. This describes what the user will do in order to perform some functions.

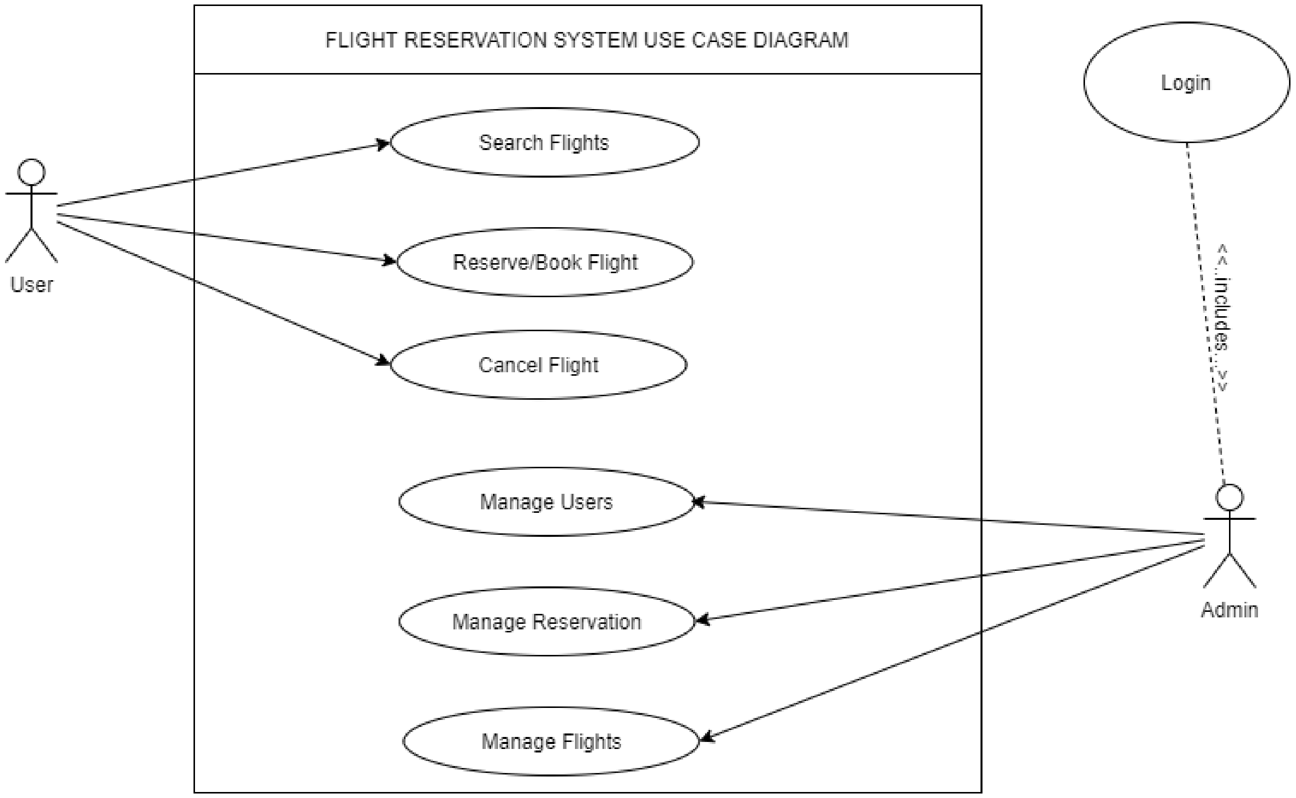
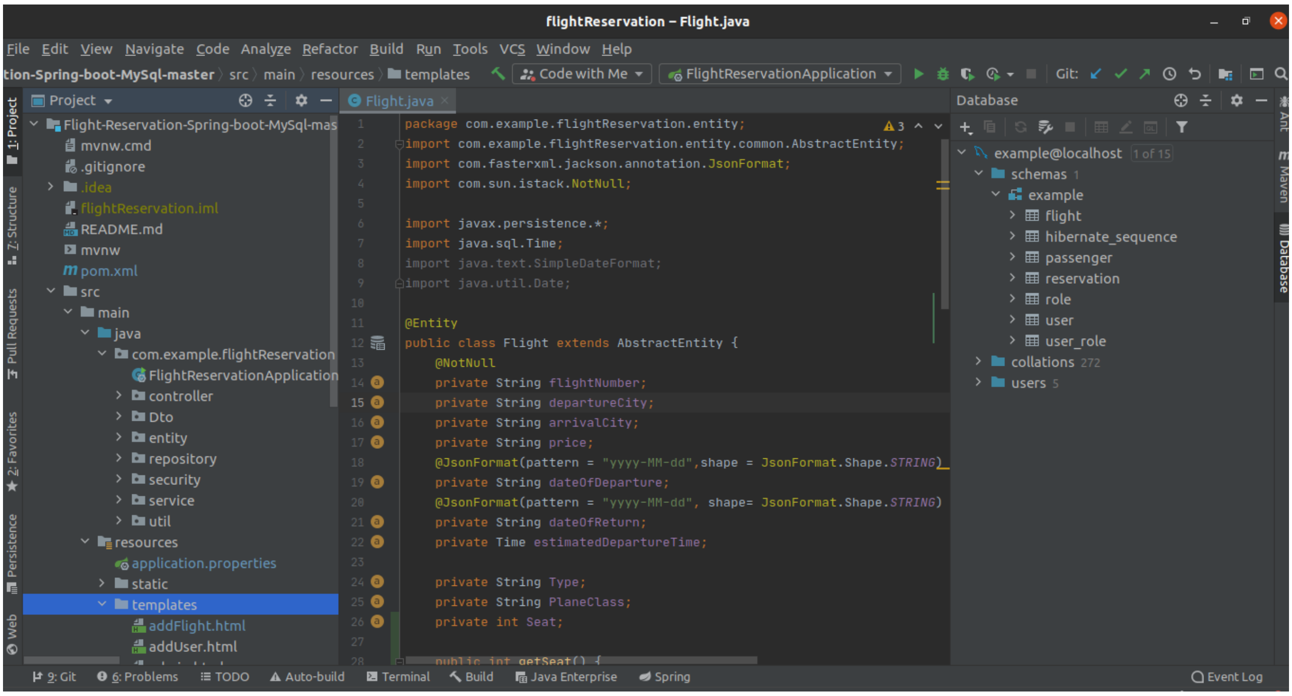


Figure 1: Use case diagram

CHAPTER TWO

2.0 PROJECT STRUCTURE

The project has a Spring MVC structure That is it Has the models/entities, views and controllers

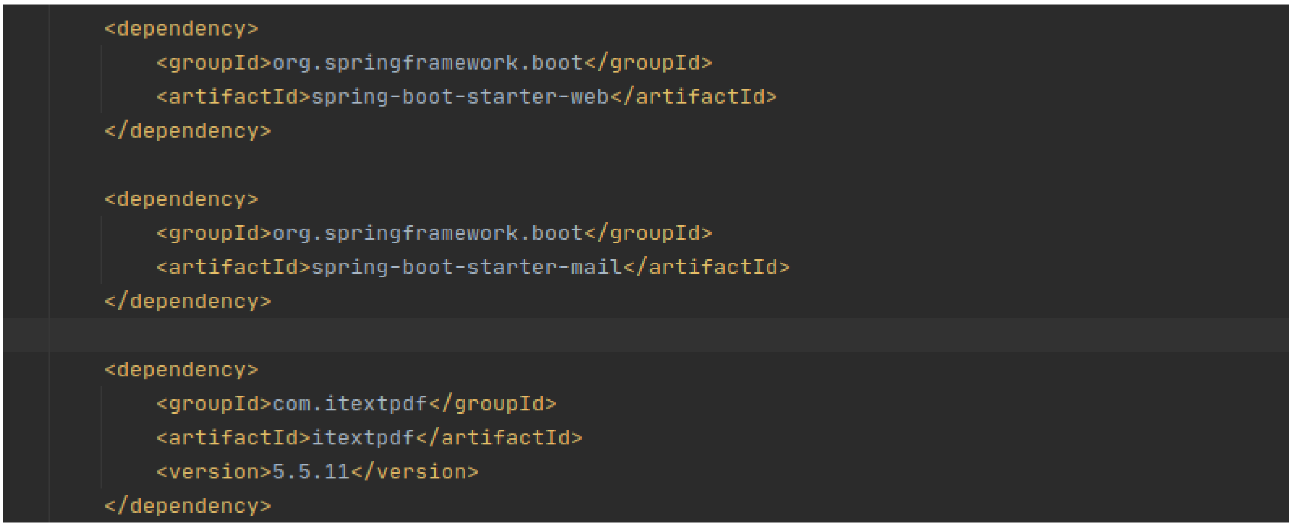


```java
package com.example.flightReservation.entity;
import com.example.flightReservation.entity.common.AbstractEntity;
import com.fasterxml.jackson.annotation.JsonFormat;
import com.sun.istack.NotNull;

import javax.persistence.*;
import java.sql.Time;
import java.text.SimpleDateFormat;
import java.util.Date;

@Entity
public class Flight extends AbstractEntity {
    @NotNull
    private String flightNumber;
    private String departureCity;
    private String arrivalCity;
    private String price;
    @JsonFormat(pattern = "yyyy-MM-dd",shape = JsonFormat.Shape.STRING)
    private String dateOfDeparture;
    @JsonFormat(pattern = "yyyy-MM-dd", shape= JsonFormat.Shape.STRING)
    private String dateOfReturn;
    private Time estimatedDepartureTime;

    private String Type;
    private String PlaneClass;
    private int Seat;

    public int getSeat() {
```

Figure 2: Project structure

2.0.1 Pom.xml

This is an Xml based file that contains all the dependencies that are used in the project. This file is known as the Configuration file

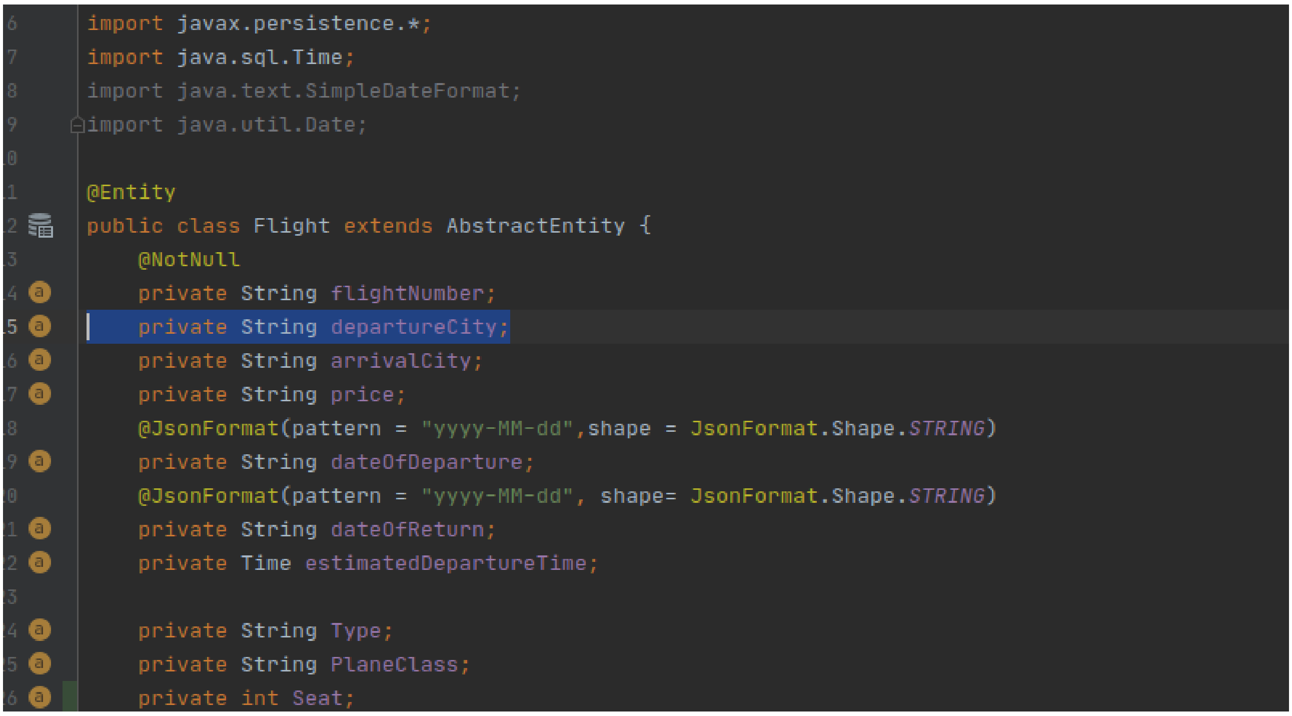


```xml
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-mail</artifactId>
    </dependency>

    <dependency>
        <groupId>com.itextpdf</groupId>
        <artifactId>itextpdf</artifactId>
        <version>5.5.11</version>
    </dependency>
```

Figure 3: Configuration File

2.0.2 Model

This is the Data Access Layer, it contains the core data of the application. The Model provides Data to the view through the controller. A model can be termed as a Table in the database, It Contains The attributes of Table and its data types. See the Figure below

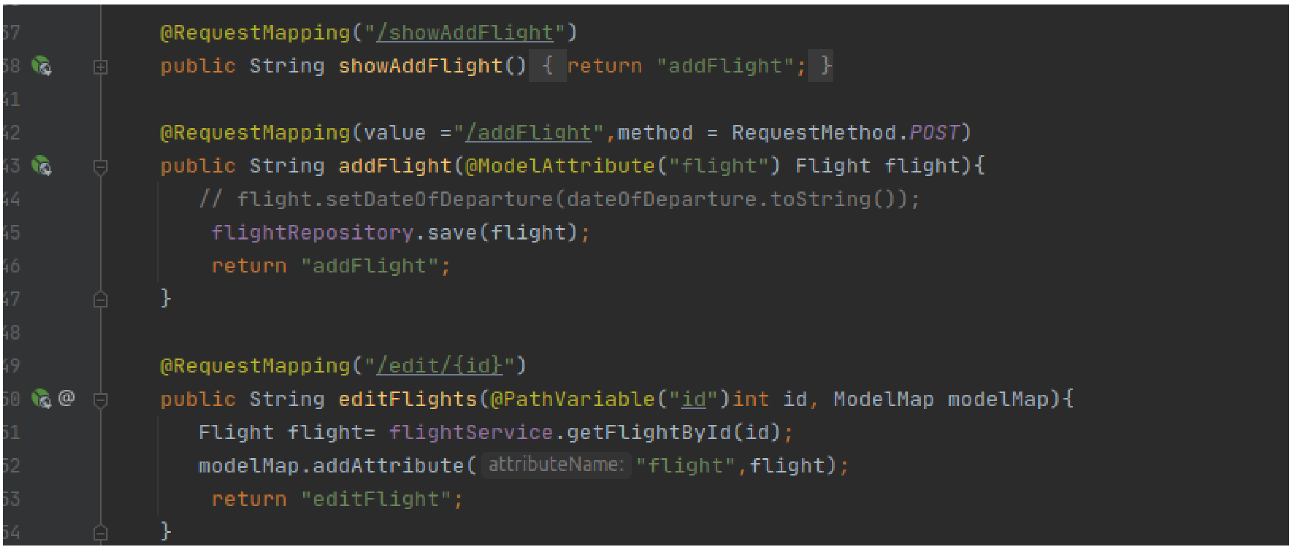


```java
import javax.persistence.*;
import java.sql.Time;
import java.text.SimpleDateFormat;
import java.util.Date;

@Entity
public class Flight extends AbstractEntity {
    @NotNull
    private String flightNumber;
    private String departureCity;
    private String arrivalCity;
    private String price;
    @JsonFormat(pattern = "yyyy-MM-dd",shape = JsonFormat.Shape.STRING)
    private String dateOfDeparture;
    @JsonFormat(pattern = "yyyy-MM-dd", shape= JsonFormat.Shape.STRING)
    private String dateOfReturn;
    private Time estimatedDepartureTime;

    private String Type;
    private String PlaneClass;
    private int Seat;
```

Figure 4: Model Class

2.0.3 Controller

The Controller is the layer that integrates the data from/to model and the view. It Contains the business logic of the application. A Controller contains Annotation which is used to map the request as Methods for example get, post, put…. See the figure below



```java
    @RequestMapping("/showAddFlight")
    public String showAddFlight() { return "addFlight"; }

    @RequestMapping(value ="/addFlight",method = RequestMethod.POST)
    public String addFlight(@ModelAttribute("flight") Flight flight){
       // flight.setDateOfDeparture(dateOfDeparture.toString());
        flightRepository.save(flight);
        return "addFlight";
    }

    @RequestMapping("/edit/{id}")
    public String editFlights(@PathVariable("id")int id, ModelMap modelMap){
       Flight flight= flightService.getFlightById(id);
       modelMap.addAttribute( attributeName: "flight",flight);
        return "editFlight";
    }
```

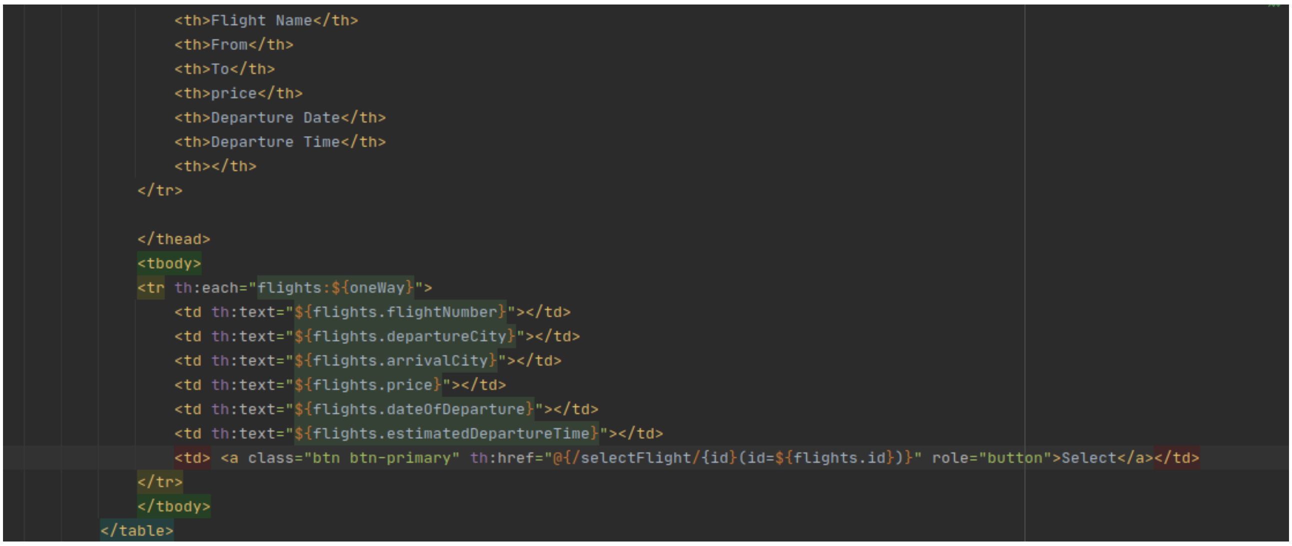
Figure 5: The Controller Class

2.0.4 View

This is the layer used to present the information from the model in a particular format. There are different Template Engines used to present information from the model to the view

- Java Server Pages(Jsp)
- Thymeleaf
- FreeMarker

In this project I used Thymeleaf Template Engine, th: shows that Thymeleaf view is used. below is an example

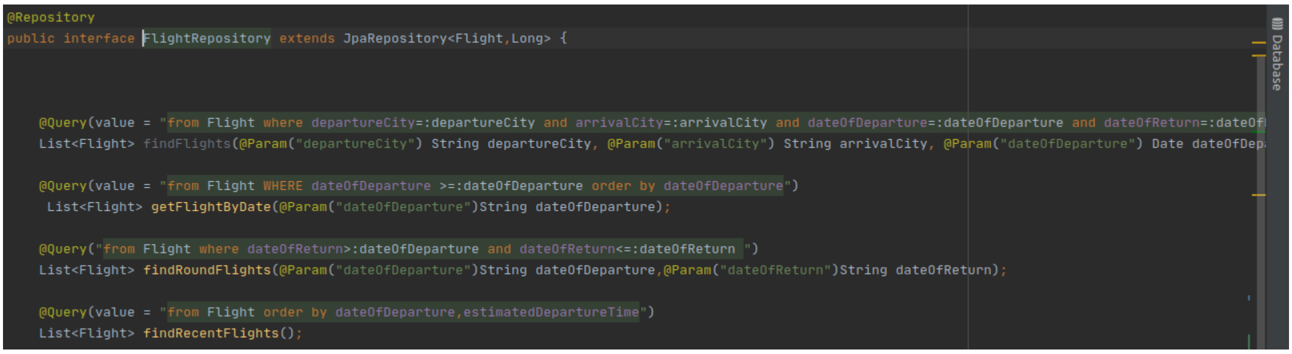


```
        <th>Flight Name</th>
        <th>From</th>
        <th>To</th>
        <th>price</th>
        <th>Departure Date</th>
        <th>Departure Time</th>
        <th></th>
    </tr>

    </thead>
    <tbody>
    <tr th:each="flights:${oneWay}">
        <td th:text="${flights.flightNumber}"></td>
        <td th:text="${flights.departureCity}"></td>
        <td th:text="${flights.arrivalCity}"></td>
        <td th:text="${flights.price}"></td>
        <td th:text="${flights.dateOfDeparture}"></td>
        <td th:text="${flights.estimatedDepartureTime}"></td>
        <td> <a class="btn btn-primary" th:href="@{/selectFlight/{id}(id=${flights.id})}" role="button">Select</a></td>
    </tr>
    </tbody>
</table>
```

Figure 6: The Thymeleaf view

2.1DATABASE DESIGN

MySQL is the Relational Database used in this project. The Spring framework provide an interface for working with Database, The Java Persistence API (API). JPQL (Java Persistence Query Language) Is the language used to write queries in the database. This Queries are written above the abstract method in the Repository. JPQL queries starts with @Query annotation as seen in the below diagram



```
@Repository
public interface FlightRepository extends JpaRepository<Flight,Long> {



    @Query(value = "from Flight where departureCity=:departureCity and arrivalCity=:arrivalCity and dateOfDeparture=:dateOfDeparture and dateOfReturn=:dateOf
    List<Flight> findFlights(@Param("departureCity") String departureCity, @Param("arrivalCity") String arrivalCity, @Param("dateOfDeparture") Date dateOfDep

    @Query(value = "from Flight WHERE dateOfDeparture >=:dateOfDeparture order by dateOfDeparture")
     List<Flight> getFlightByDate(@Param("dateOfDeparture")String dateOfDeparture);

    @Query("from Flight where dateOfReturn>:dateOfDeparture and dateOfReturn<=:dateOfReturn ")
    List<Flight> findRoundFlights(@Param("dateOfDeparture")String dateOfDeparture,@Param("dateOfReturn")String dateOfReturn);

    @Query(value = "from Flight order by dateOfDeparture,estimatedDepartureTime")
    List<Flight> findRecentFlights();
```

Figure 7: The Repository Interface

The abstract methods above are used by the controllers for movement of data across the database.

2.1.1 ENTITY RELATIONSHIP DIAGRAM

Is a type of structural diagram for use in database design. An ERD contains different symbols and connectors that visualize two important information The major entities within the system scope, and the inter-relationships among these entities.

The Database has six tables

1. Flight
2. Passenger
3. Reservation
4. User
5. Role
6. User_role

All the six tables are well normalized and hence no redundancy. The following diagram shows the database Schema.

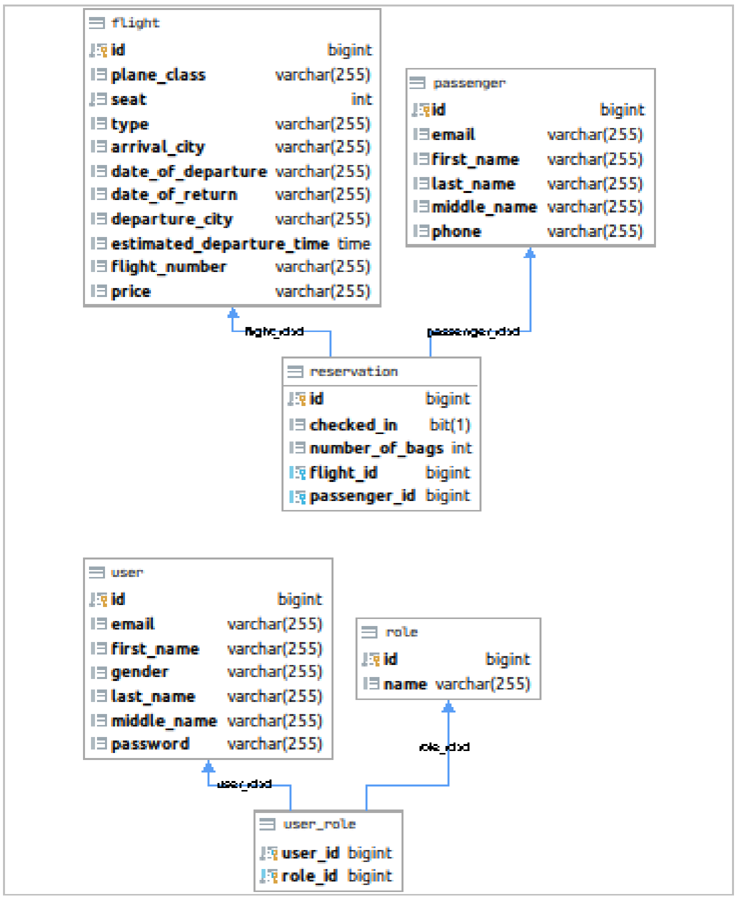


Figure 8: E-R Diagram

2.2USER INTERFACE

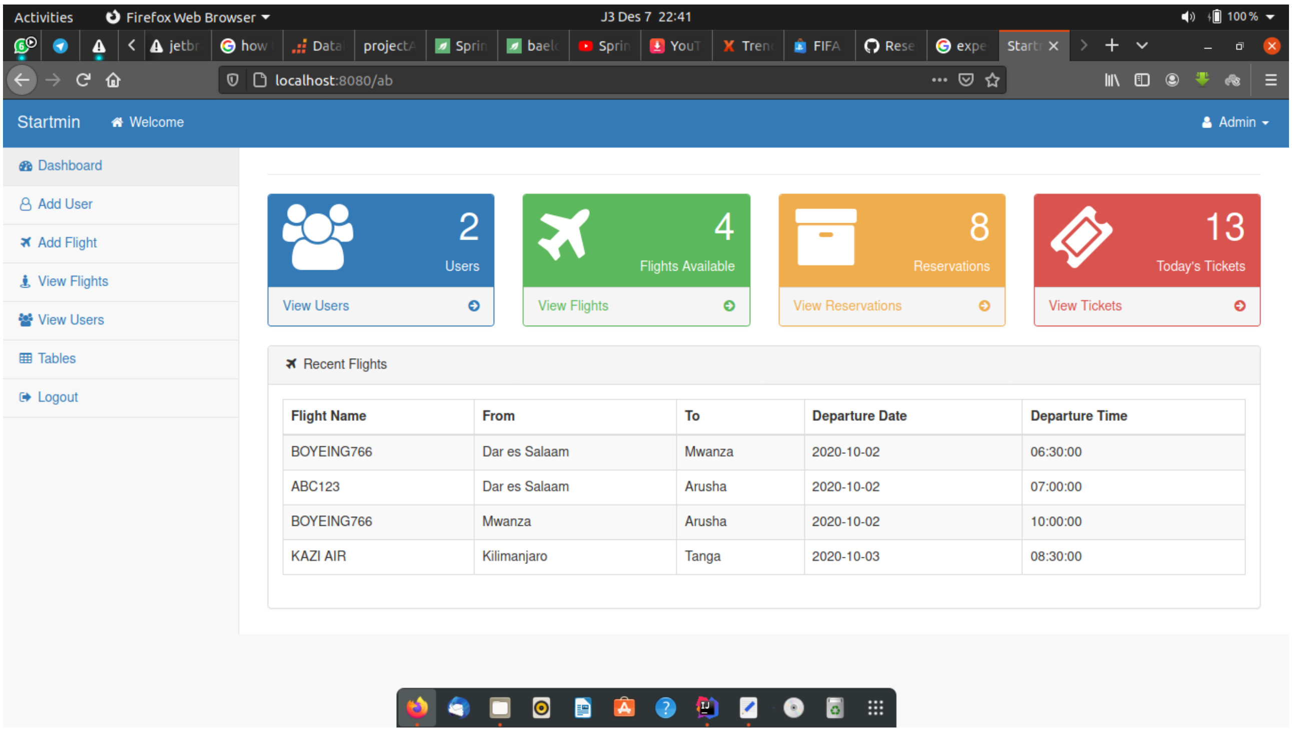
The UI in this project is divided into two parts

1) ADMIN INTERFACE
2) USER INTERFACE

2.2.1 ADMIN INTERFACE

The Admin is known as the Super User. Has the ability to Read, Update or delete user's data.

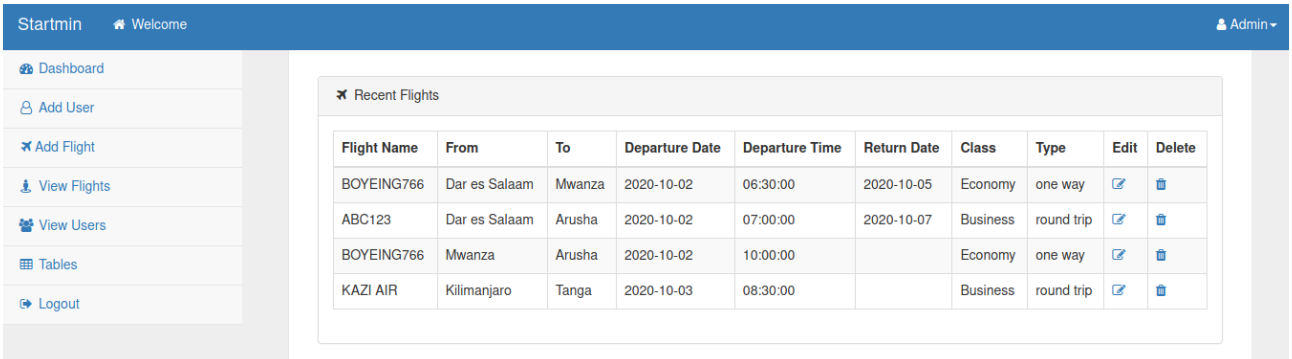
The Admin interface has a dashboard that shows useful information. The Admin is able to create a user and assign a role to that user, edit/delete user details, add/edit flight information and many other functionalities.



Flight Name	From	To	Departure Date	Departure Time
BOYEING766	Dar es Salaam	Mwanza	2020-10-02	06:30:00
ABC123	Dar es Salaam	Arusha	2020-10-02	07:00:00
BOYEING766	Mwanza	Arusha	2020-10-02	10:00:00
KAZI AIR	Kilimanjaro	Tanga	2020-10-03	08:30:00

Figure 9: Admin Dashboard

The Admin Is able to view all the flight details and modify when there are any changes as seen In the below diagram



Flight Name	From	To	Departure Date	Departure Time	Return Date	Class	Type	Edit	Delete
BOYEING766	Dar es Salaam	Mwanza	2020-10-02	06:30:00	2020-10-05	Economy	one way	✎	🗑
ABC123	Dar es Salaam	Arusha	2020-10-02	07:00:00	2020-10-07	Business	round trip	✎	🗑
BOYEING766	Mwanza	Arusha	2020-10-02	10:00:00		Economy	one way	✎	🗑
KAZI AIR	Kilimanjaro	Tanga	2020-10-03	08:30:00		Business	round trip	✎	🗑

Figure 10: Flight Details

This is the User login page. Only Authorized users are allowed to login. Example the Admin

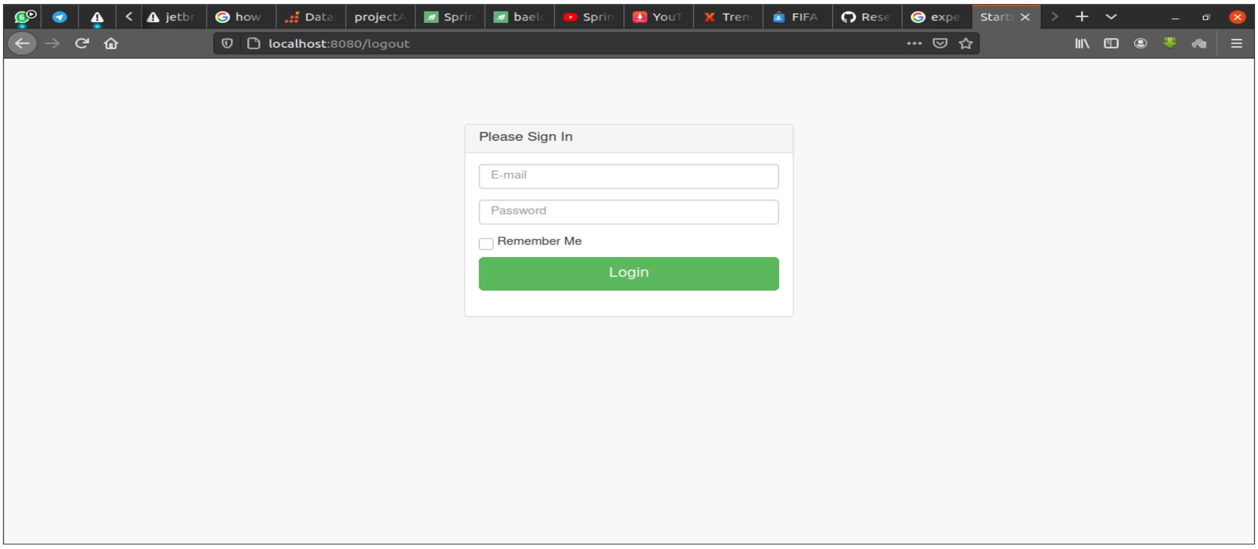


Figure 11: User Login Page

2.2.2 USER INTERFACE

The User interface is so simple and user friendly. A user searches his/her route(from/to) and he/she sees all the available flights at that route according to departure time. After he selects a flight he provides his necessary information, names and email, as his ticket will be sent through the email.

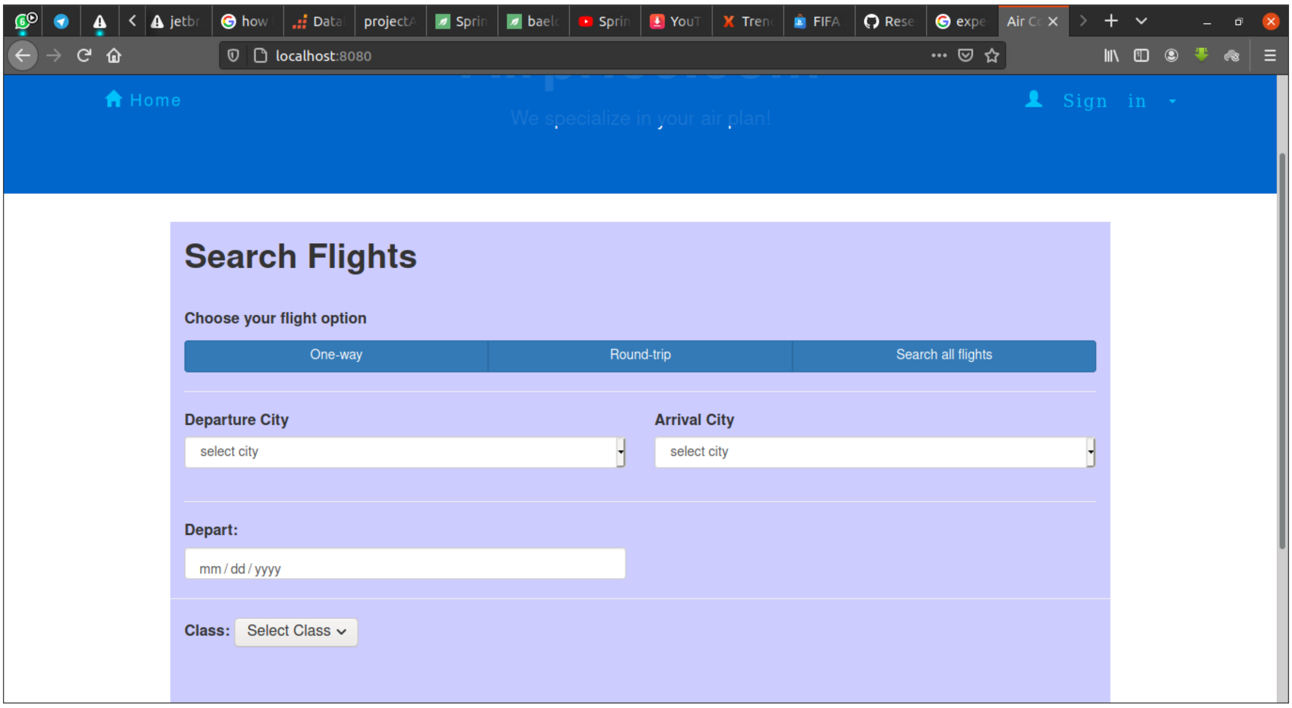
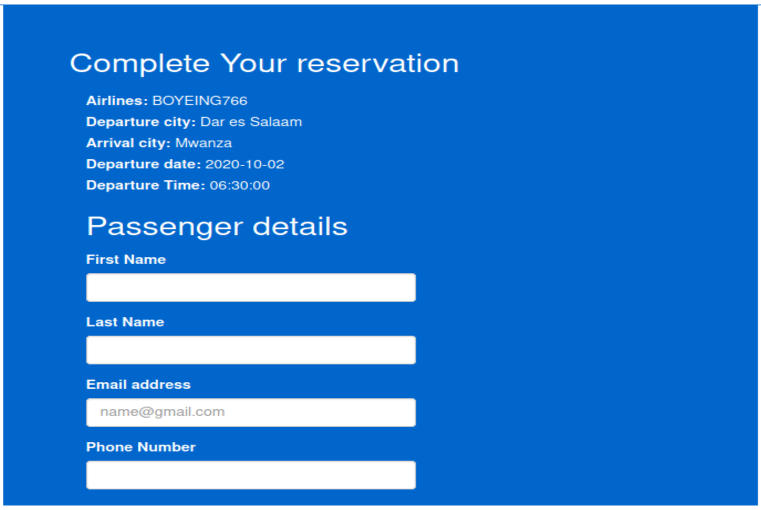


Figure 12: Homepage

When a User selects a flight he/she is supposed to provide his information as portrayed in the figure below. After submission the user will receive an email address.



Complete Your reservation

Airlines: BOYEING766
Departure city: Dar es Salaam
Arrival city: Mwanza
Departure date: 2020-10-02
Departure Time: 06:30:00

Passenger details

First Name

Last Name

Email address: name@gmail.com

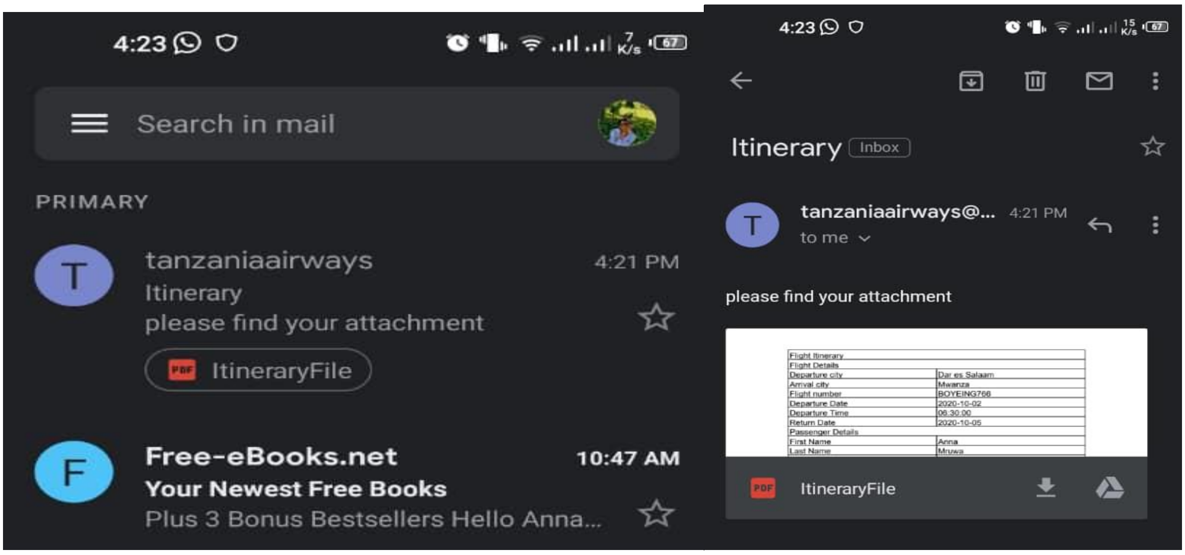
Phone Number

Figure 13: Reservation

2.3EMAIL UTILITY

Email Utility is an Interface that is used to support email sending functionality. This Interface is implemented by the java Class EmailSenderImpl which uses Email Sender API to send out emails. In this project Gmail is used as the email host.

In the Configuration file(pom.xml) we add the Mail dependency and then create an email address that will be used to deliver emails to clients.



tanzaniaairways 4:21 PM
Itinerary
please find your attachment
ItineraryFile

Free-eBooks.net 10:47 AM
Your Newest Free Books
Plus 3 Bonus Bestsellers Hello Anna...

tanzaniaairways@... 4:21 PM
to me

please find your attachment

Flight Itinerary	
Flight Details	
Departure city	Dar es Salaam
Arrival city	Mwanza
Flight number	BOYEING766
Departure Date	2020-10-02
Departure Time	06:30:00
Return Date	2020-10-05
Passenger Details	
First Name	Anna
Last Name	Mruwa

ItineraryFile

Figure 14: Email demonstration

2.4DOCUMENT GENERATION

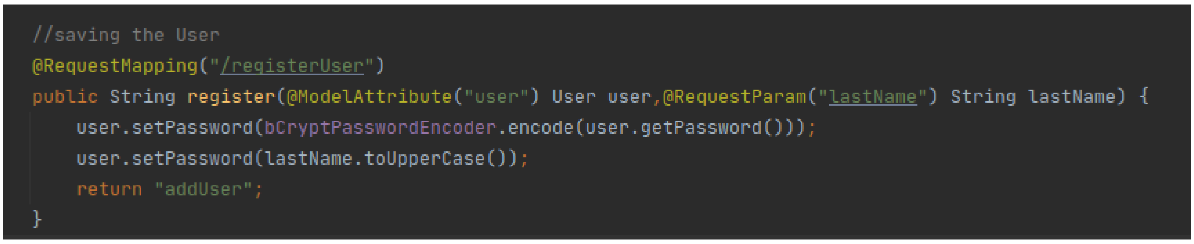
Document Generation is one of the futures in this project, The Application is able to Generate a document that contains flight and passenger details and instructions on how the passenger should pay for the ticket. The document is sent as a pdf to the passenger email.

2.5 SECURITY

Security is an Important feature any application should have, because it keeps safe the website from intruders/hackers who steal data in systems.

Spring framework has Spring Security that is used in protecting the application. Spring Security is a powerful and highly customizable authentication and access-control framework. Spring Security focuses on providing both authentication and authorization to Java applications.

The features of Spring Security enables to encode passwords. When a user is registered the password is captured and encoded in the database.



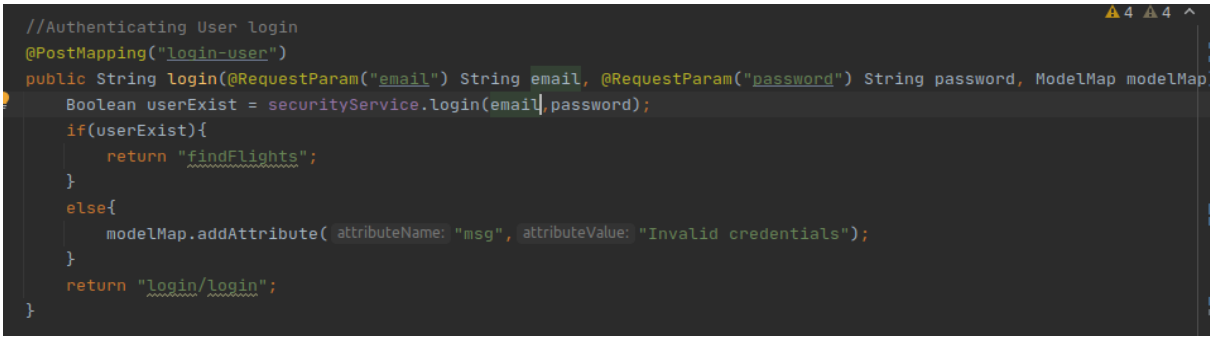
```
//saving the User
@RequestMapping("/registerUser")
public String register(@ModelAttribute("user") User user,@RequestParam("lastName") String lastName) {
    user.setPassword(bCryptPasswordEncoder.encode(user.getPassword()));
    user.setPassword(lastName.toUpperCase());
    return "addUser";
}
```

Figure 15: Password Encoding

2.5.1 AUTHENTICATION

Authentication Is the process of recognizing user's identity. This mechanism is associated with identifying user's credentials like username and password.

In the diagram below user credentials are captured and validated by the spring security service, if the data are missing in the database or the credentials are not correct a user is notified to enter again.



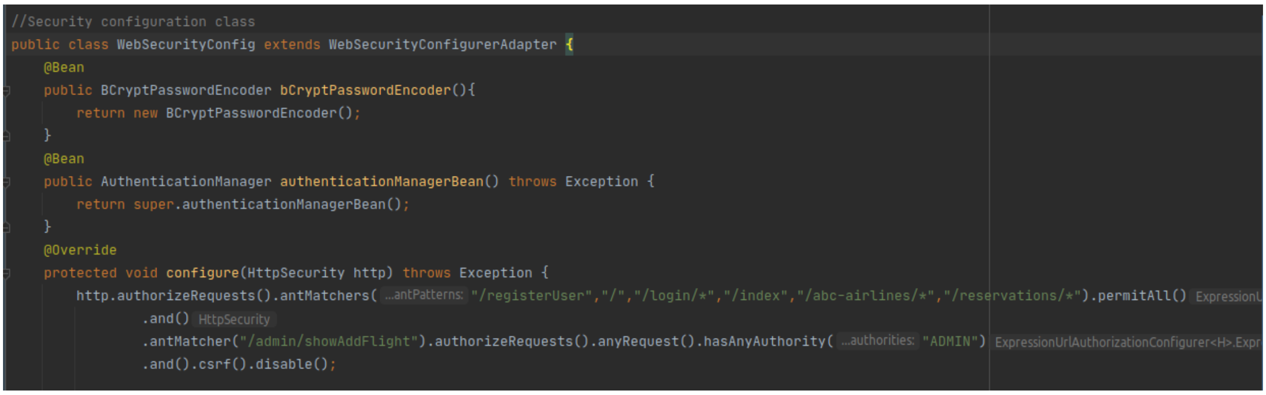
```
//Authenticating User login
@PostMapping("login-user")
public String login(@RequestParam("email") String email, @RequestParam("password") String password, ModelMap modelMap
    Boolean userExist = securityService.login(email,password);
    if(userExist){
        return "findFlights";
    }
    else{
        modelMap.addAttribute( attributeName: "msg", attributeValue: "Invalid credentials");
    }
    return "login/login";
}
```

Figure 16: User Authentication

2.5.2AUTHORIZATION

Authorization is a security mechanism to determine access levels or user/client privileges related to system resources including files, services, computer programs, data and application features. This is the process of granting or denying access to a network resource which allows the user access to various resources based on the user's identity.

In the below code a User is granted a role as an ADMIN and given access privileges to access all the html pages and the all the functionalities. The Admin is a super User



```
//Security configuration class
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
    @Bean
    public BCryptPasswordEncoder bCryptPasswordEncoder(){
        return new BCryptPasswordEncoder();
    }
    @Bean
    public AuthenticationManager authenticationManagerBean() throws Exception {
        return super.authenticationManagerBean();
    }
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests().antMatchers( …antPatterns: "/registerUser","/","/login/*","/index","/abc-airlines/*","/reservations/*").permitAll() ExpressionU
                .and() HttpSecurity
                .antMatcher("/admin/showAddFlight").authorizeRequests().anyRequest().hasAnyAuthority( …authorities: "ADMIN") ExpressionUrlAuthorizationConfigurer<H>.Expr
                .and().csrf().disable();
```

Figure 17: User Authorization

2.6 CONCLUSIONS AND RECOMMENDATIONS

2.6.1 CONCLUSION

Accordingly, practical training is imperative, for it is of paramount to both students and to the college as a whole. It enhances students to learn and understand many practical training programs that are not covered in colleges and provides the ground for students to strengthen the theoretical knowledge obtained in classes into practice necessary for their carriers therefore, students can integrate their theory into practice.

The training exposes students to new challenges and enables students to experience different working environments before they are actually employed upon completion of their studies.

Broadly, I enjoyed working with **ICT PACK** simply because I managed to meet almost all my expectations. Most of problems I encountered were solved on time and so passionately, the working environment was very good, admirable and favorable to some extend for supporters, and me, as the staff members were so cooperative, kindly to me.

I would like also to extend my thanks to ICT PACK for their decision of providing places for a great number of students and their assistance in practical training. I myself appreciate for their conscience.

Moreover; through practical training firms are able to receive new ideas, get flexible and adaptable labor forces that are eager to work well in teams and do also build closer links with colleges and universities.

2.6.2 RECOMMENDATIONS

In fact, practical training is the most interesting course to students because it is simpler to grasp things practically than by using theory itself. Therefore, theoretical knowledge has to be supplemented by practice to improve understanding of material facts learned in classes.

REFERENCES

Craig Walls (2014), *Spring Boot In Action*

https://spring.io/

https://www.baeldung.com/